# 8.1 Knowledge Organiser

| | |
|---|---|
| **Coding Scheme:** | A set standard to help understand code |
| **Binary:** | base-2 number system invented that is made up of only two numbers: 0 and 1. |
| **Bit:** | A single Binary Unit ( 1 or 0) |
| **Byte**: | 8 Bits Digits (10101010) |
| **ASCII:** | Used in computers for encoding characters. It employs the binary digits 0 and 1. In its original form, each character corresponds to 7 bits. |

| Unit | Capacity |
|---|---|
| Bit | 1 / 0 |
| Byte | 8 Bits (e.g. 01010101 ) |
| Kilobyte **(Kb)** | 1000 Bytes |
| MegaByte **(Mb)** | 1000 Kilobytes |
| GigaBytes **(Gb)** | 1000 MegaBytes |
| TeraByte **(Tb)** | 1000 Gigabytes |

/ As the file size increases divide by 1000 as it decreases Multiply by 1000. X

| **Converting between Units** | | |
|---|---|---|
| Convert 5Tb to Gb = | **5Tb x 1000 =** | **5000GB** |
| Convert 500Kb to Mb = | **500Kb / 1000 =** | **0.5 Mb** |
| Try doing these | | |
| Convert 500Mb to Kilobytes | **500Mb x 1000=** | |
| Convert 10Gb to Terabytes | **10Gb / 1000=** | |

## Decimal to Binary

To convert 70 to binary place a 1 under each number needed to make the number 70 starting at the largest first.

For Example:

Put a 1 under 64, then

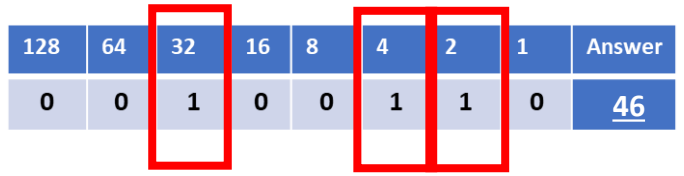
70-64 = 6 So put a 1 under 4 and a 1 under 2 which makes 6.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Answer |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 70 |
| | 6 | | | | 2 | 0 | | |

## Binary to Decimal

To convert 00100110 place a 1 under each number, (Go from right to left).

Add all the 1's together 32+8+4+2=46.



| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Answer |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 46 |

## Try converting these decimal number to binary

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 15 |

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 81 |

## Try converting this binary to decimal

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 0 | 0 | 1 | 1 | 1 | |

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 0 | 1 | 0 | 1 | 1 | |

Binary addition Rules

Binary addition follows simple rules similar to decimal addition, but since binary numbers only use two digits, 0 and 1, there are only a few possible outcomes for each sum. Here are the rules:

0 + 0 = 0

0 + 1 = 1

1 + 1 = 10 (0 carry 1)

1 + 1 + 1 = 11 (1 carry 1)

| Character | Binary Representation | Number of bits | Number of bytes |
|---|---|---|---|
| A | 01000000 | 8 | |
| A B | 11000010 10100011 | 16 | |
| A B C | 11100010 10000001 10000010 | 24 | |
| A B C D | 11110000 10011101 10000100 10011110 | 32 | |

# 8.1 Data Representation

## What I need to know:

| | | | |
|---|---|---|---|
| **Data sizes** | | | |
| Define the term bit. | | | |
| How many bits are in a byte? | | | |
| I can name the file sizes and put them into order | | | |
| I know how to convert between the file types by multiplying or dividing by 1000 | | | |
| **Binary** | | | |
| I know what the two binary number are | | | |
| I can apply Binary logic to other applications | | | |
| Order the Binary units from smallest to largest. | | | |
| I know how to make a binary table of numbers | | | |
| I can convert binary numbers to a decimal number | | | |
| I can convert a number into Binary | | | |
| I can use the 4 rules of binary addition to correctly add 2 binary numbers together | | | |